International Journal of Computer Applications Technology and Research
Volume 11–Issue 11, 351-358, 2022, ISSN:-2319–8656
DOI:10.7753/IJCATR1111.1001

Control of Arm Robot Based on Finger Motion

Salman Bintang
Faculty of Engineering
Universitas Negeri Medan
Medan, Indonesia

Ressy Dwitias Sari
Faculty of Engineering
Universitas Negeri Medan
Medan, Indonesia

Bakti Dwi Waluyo
Faculty of Engineering
Universitas Negeri Medan
Medan, Indonesia

Abstract: Robotics is a constantly evolving field nowadays. A robot is a mechanical device that can perform physical tasks while being controlled and supervised by humans. Several robots have been created to perform risky tasks impossible for humans to act directly. The robot arm is one of the most popular types. In this project, it is proposed to control a robot arm using finger and hand movements. However, the traditional way of gesture control employs sensor-based gloves to track motion, which is time and energy-intensive due to its weight. To overcome this problem, using vision for motion tracking can be the most suited method at hand currently. The ML-based computer vision method provides real-time landmarks on different hand points. With these landmarks, we can calculate various mathematical parameters that can be used to control the robot arm. Movements that are landmarked are processed and formulated using mathematical parameters. The final output of mathematical parameters is fed to the microcontroller. Signals from the microcontroller will be the desired input signal to the motors. The rotation of the robot arm is controlled using the movement of the human arm. This project’s end application is like human operation in dull, dirty, and dangerous environments. The project’s primary goal is to protect those who come into direct contact with these environments. Bomb defusing, painting, welding, and other applications are a few examples of where this project is helpful.

Keywords: robot arm; media pipe; gesture; opencv; arduino; servo

1. INTRODUCTION

People nowadays have a constant need for extra assistance systems to improve productivity and safety. This will necessitate the use of automation systems. Standard automation systems and skilled and well-trained employees are required to produce high-quality products. With the advancement of technology, these machines that formerly required human assistance to work have been made to operate spontaneously without the need for human power. Robots are one of the most common components of automation systems. Robot arms are machines that are configured to do a given activity as well as duty fastly, effectively, but also precisely. They’re generally motor actuated and used for doing heavy or highly repetitive operations quickly and consistently over long durations. They’re instrumental in the machining, manufacturing, production and assembling industries. A standard industrial robot arm consists of manipulators, articulations and a series of joints that works with each other to mimic the motion and capabilities of a human arm as precisely as feasible. A gesture-controlled robot arm is a sort of action that functions by detecting and responding to hand gestures by robots. Gesture-based computing interfaces where the human body interacts with digital content, such as computer vision-based approaches, and a gesture-controlled robot arm is a type of robot that works based on signals supplied by hand gestures. The robotic arm moves and executes the work based on human hand gestures, and this technology mimics human hand motions.

A vision-based method makes it simple and easy to manage a robot arm for various operations such as palletizing, painting, welding, etc. Articulated robots are those that have rotating joints. Axes are the term used in robotics to describe these joints. Servo motors are commonly used to power articulated robots. Servo motors can be two-axis design or up to ten axes or more. In industrial robots, four to six axes are joined. In industrial applications, six axes are the most typical.

2. RELATED WORK

Much research has been conducted in the area related to Gesture-Based Robot Arms Control Poltak Sihombing et al. [1] focus on the robot arm control is based on the movement of fingers and hands. The research aims to briefly describe the possession of a simple robot arm by the action of human fingers and hands. This project has created a robot arm using a Fuzzy logic technique to control the movement of the robot arm. Yanmin Zhu et al. [2] focus on basic background subtraction, skin color detection, hand reach detection, and palm detection. Detecting a full palm is used to determine whether the hand extends beyond the camera’s range of vision. The ergonomics idea assesses whether the pointer is outside the camera’s field of view.

There has also been some more study done in this area. Using RGB cameras, Pramod Kumar et al. [3] investigated numeric and subjective comparisons of algorithms and approaches. A collection of 13 metrics developed from various algorithm attributes and the experimental methodologies used in algorithm assessment are utilized to evaluate algorithms quantitatively. The project emphasizes the need to consider these factors and the algorithm’s identification accuracy when predicting its success in real-world applications. Sakshi Sharma et al. [4] described how he used accelerometers to control five servo motors with gestures. The use of an accelerometer to drive a robot arm is explored and implemented in the hardware [5].

3. SYSTEM COMPONENTS

In recent years, technology has provided various sources of essential items that assist people in meeting their requirements. As a result, the software and algorithms help developers use the package, either an open-source platform or a commercial product, to build their goods, which is helpful for various industries. Such software and algorithms will be updated periodically to make a better component. The following software and algorithms are used in this project. They are PyCharm Edu 2021.1.2-Software, Arduino IDE-Software, Machine Learning Pipeline-Algorithm, and

MediaPipe–Library. The following components are used in this project. They are Arduino Uno, Servo motors, and a 5V Battery.

3.1 PyCharm IDE

PyCharm is an integrated development environment (IDE) for computer programming that focuses on Programming in Python. It was created by the JetBrains Corporation (previously identified as IntelliJ). PyCharm IDE supports website design and data science with code analyzing, visual debugging, an integrated unit tester, version control system integration, and Django and Anaconda integration. PyCharm is compatible with Linux, Mac OS X, and Windows versions. PyCharm includes coding aid and inspection, auto-completion, syntax and error highlights, linter integration, and rapid fixes.

3.2 Arduino IDE

The Arduino Integrated Development Environment (IDE) is a software application that functions on Windows, Mac OS X, and Linux computers. It's used to compile and execute programs on Arduino boards and other development boards that enable third-party cores. Connecting to the Arduino and Genuino devices and uploading code communicates with them. The Arduino IDE includes a text editor, a serial monitor, a text editor, a toolbar with activity buttons, and several menus.

3.3 Machine Learning Pipeline

A machine learning pipeline codifies and automates the process of developing a machine learning model. Data extraction, pre-processing, model training, and deployment are all handled by machine learning pipelines. MediaPipe Hands uses a machine-learning channel comprising several interconnected models. A palm detection model which works on the entire frame comes back as an aligned hand boundary. A hand landmark model extracts high-fidelity 3-dimensional hand key points from the palm detector-cropped picture area. Moreover, crops may be made in the pipelines using the hand landmark identified in the previous frame. Palm detection is only required to relocate the hand when the landmark models no longer detect its presence.

3.4 MediaPipe

Recognizing the motion and form of hands could enhance the user experience in various technical areas and interfaces. In augmented reality, it enables the texture of digital material and information on top of the natural world, the interpretation of sign language and the control of hand movements. Because palms frequently entirely cover themselves or one another (for instance, hand occlusions and handshaking), and there are no high contrast styles, hand perception in real-time is a challenging computer vision task. MediaPipe's Hands is a high-resolution tracking device for hands and fingers. 21 3-dimensional landmarks of a hand are determined using machine learning. A single image is used to select 21 3D landmarks of a hand using machine learning. While most current framework systems rely on complex desktop environments for inference, our approach works in real-time on a cell phone and can be scaled to many hands. By making these hand recognition abilities open to the rest of the research and innovation community, the designers expect that new applications and research routes will emerge, leading to new applications and research.

3.5 Arduino Uno

The ATmega328P is used in the Arduino Uno, created by Arduino. Cc. There are numerous digital and analog I/O pins. This allows the Arduino to be connected to external circuits. As an output, this board can run and control relays, LEDs, servos, and motors and connect to other Microcontrollers. The board has six analog and 14 digital input/output ports (six of which are utilized for PWM output. The Arduino Uno is powered by a 9 V battery or a USB wire. It can handle voltages ranging from 7 to 20 volts. The Arduino Uno is shown in fig. 1.

Figure 1. Arduino uno

3.6 Servo Motors

A servo motor is a rotary actuator that enables accurate angular position control. A motor and a position feedback sensor are used in it. To complete the system, a servo drive is necessary. The purpose is to use the feedback sensor to regulate the motor's rotational position accurately. Fig. 2 depicts a DC Servo motor.

Figure 2. Servo motor (DC)

3.7 Power Supply

A power supply is an electrical device that stores and transmits electrical energy to convert it into various forms of energy. A power supply or Battery delivers the necessary electric energy to power the load at the needed voltage, current, and frequency. The Battery is shown in fig 3.

International Journal of Computer Applications Technology and Research
Volume 11–Issue 11, 351-358, 2022, ISSN:-2319–8656
DOI:10.7753/IJCATR1111.1001

Figure 3. 5 Volt, 1500 mAH Battery

4. METHOD

A systematic methodology creates a fully functional human following load carrier in mind. This project takes a top-down, decentralized approach. There are numerous stages to the project. Step-by-step procedures were followed, starting with various literature surveys, and then questioning their concerns. After determining the most common issues, the problem statement was written. A solution was suggested for the cases above, and the best solution was chosen. The central concept for the design has been selected, and numerous existing publications and patents have been evaluated.

A significant concept was raised following an examination of the numerous publications. As a result, a conceptual design with estimated parameters was developed. After the necessary software and electronics were selected, a detailed study was conducted, and accurate parameters were established. After that, the detailed design was modeled and used to create prototypes. Finally, the prototype was tested using hand gestures to produce the desired response in physical environments. Fig. 4 depicts the methodology that was developed.

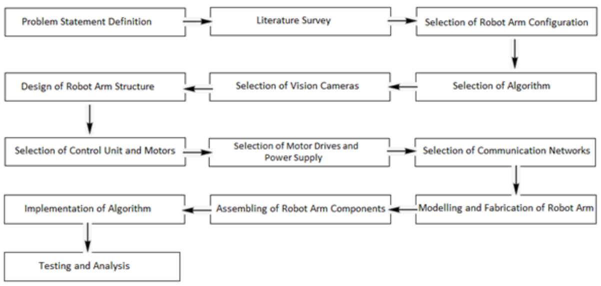

Figure 4. Process flowchart

4.1 Conceptual Design

The project is primarily concerned with software architecture, with the software being implemented by interfacing it with hardware for physical testing and analysis. The algorithm's creation and implementation may control any articulated robot for any industrial application. The programming languages utilized are Python and embedded C. The Arduino Uno is the microcontroller that operates and manages the actuators. The actuators used in work are DC Servo motors. A web camera captures the image, which is then processed using a pre-trained machine-learning pipeline. The capturing and processing program is written in PyCharm (Python IDE) and is used to acquire data from the image and send it to Arduino. The servo motor is commanded to make the desired gestures. The block diagram of the proposed model of this project is mentioned in Fig. 5.

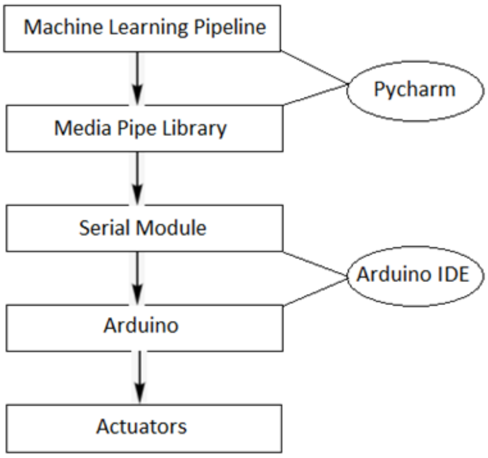

Figure 5. Block diagram of the proposed model

4.2 Detailed Design

In gesture-based robot arm control, computer vision controls the hardware. As a result, the project focuses primarily on software design, with visual testing conducted using hardware via triggering motors with required motion.

A single-shot detection model for mobile real-time use is built to recognize the beginning postures of hands, comparable to the facial recognition model in MediaPipe Face Mesh. Hand identification is a complex problem since the model must distinguish between occluded and self-occluded hands despite operating with a diverse size range of hands and a wide scale time frame for a captured image.

Regression is utilized to accomplish the exact localized key point of 21 3D hand-knuckle points inside the identified hand regions, i.e., direct point predictions, after palm identification across the whole picture.

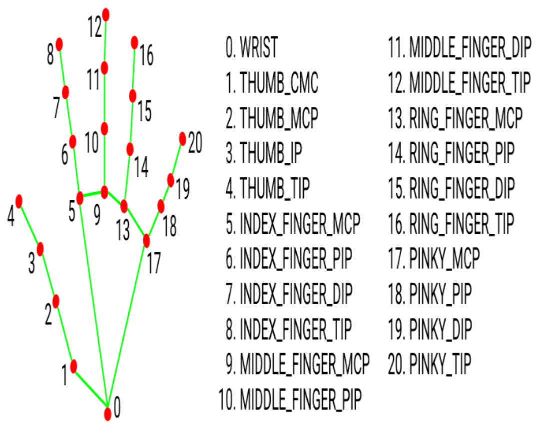

Figure 7. Hand landmarks

To get ground truth data 30 thousand actual pictures were manually tagged with 21 points in 3 dimensional coordinates, as shown in Fig. 6. The best quality artificial hand model was created across different backdrops and mapped to the appropriate three-dimensional coordinates to cover the possible hand postures better and provide more oversight on the nature of hand geometry. The 21-point hand landmarks are shown in Fig.7.

The media pipe hands algorithm analyzes real-time images and determines which palm belongs to which hand. 21-point coordinates are mapped after palm detection. The top three coordinate points are taken in all fingers from the tip to the bottom, a line is drawn connecting the three spots, and the angle between the three points is measured.

The five-finger angle is recorded in the list values and communicated to the Arduino via serial communication. According to our gesture, the Arduino controls the servo motor in real-time. The control circuit and actuators are connected as per circuit design. The course is designed in TinkerCad software, as shown in Fig.8.

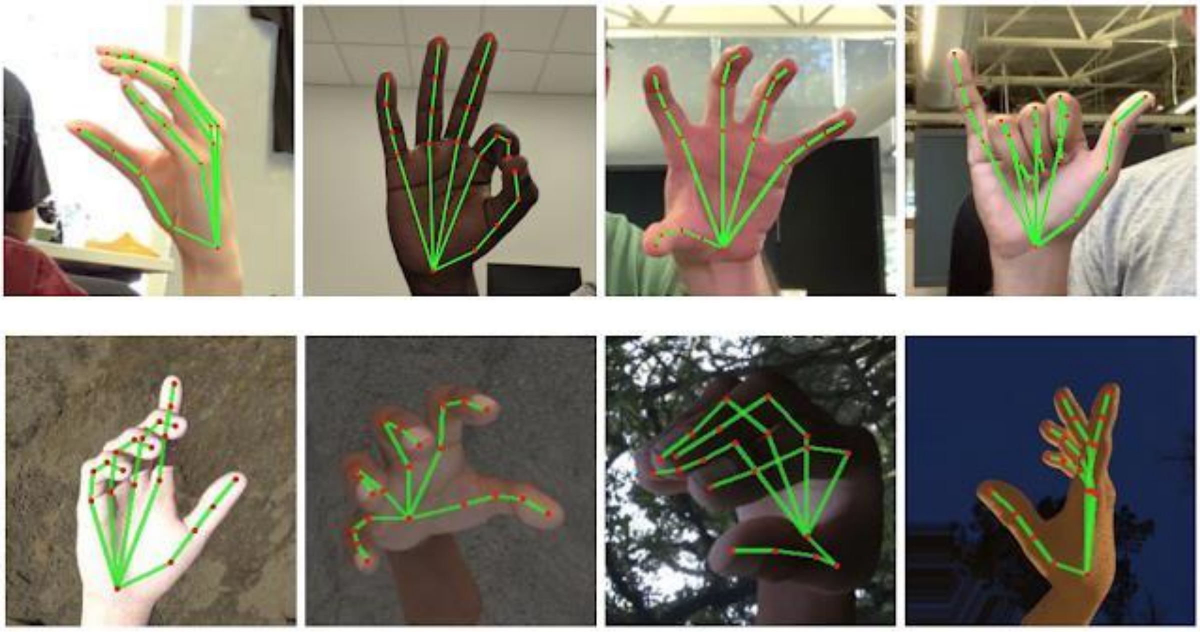

Figure 6. Various hand tracking coordinated

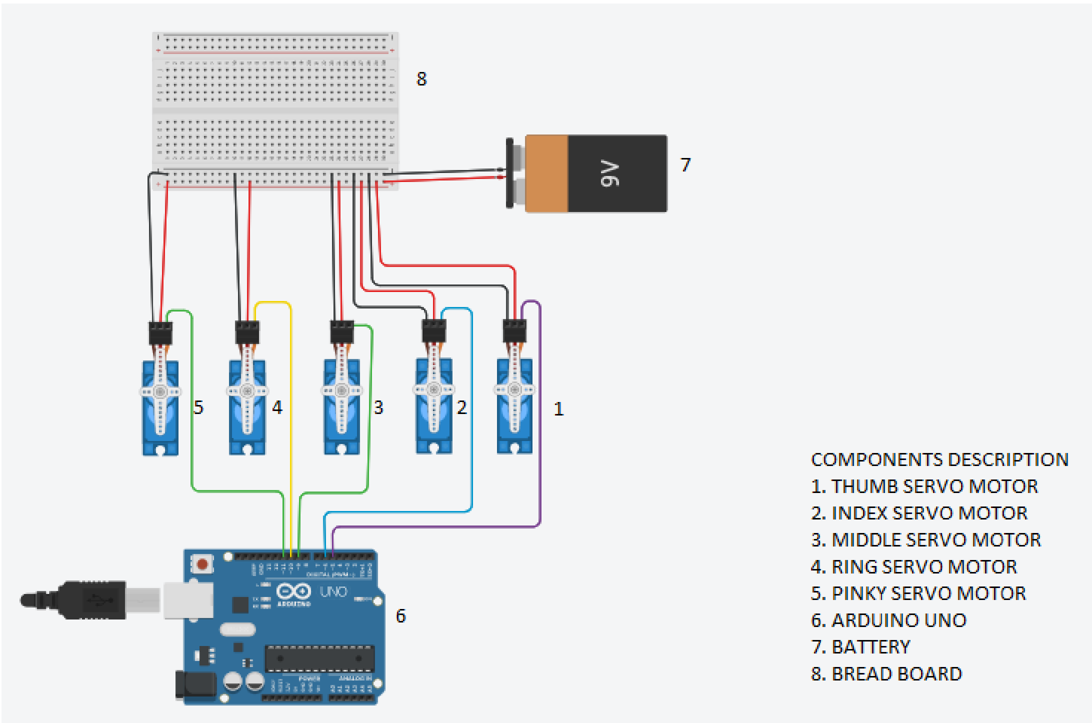

Figure 8. Circuit design

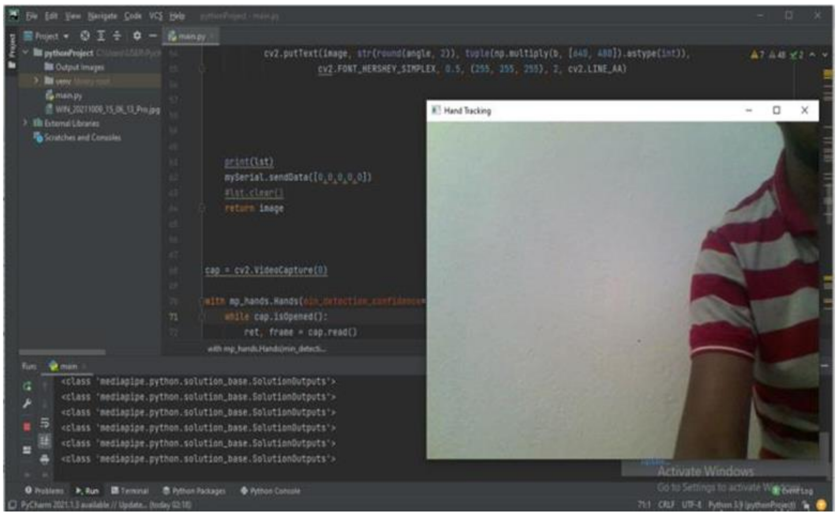

Figure 9. Video capturing

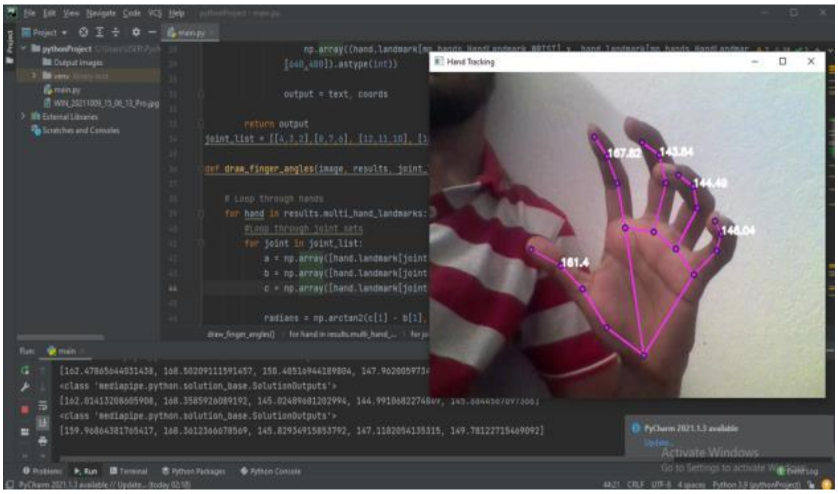

Figure 10. Processing the real-time video

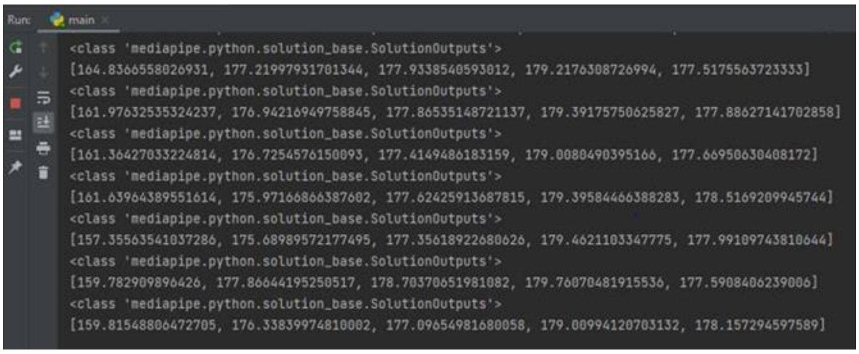

Figure 11. List Values sent to serial communications

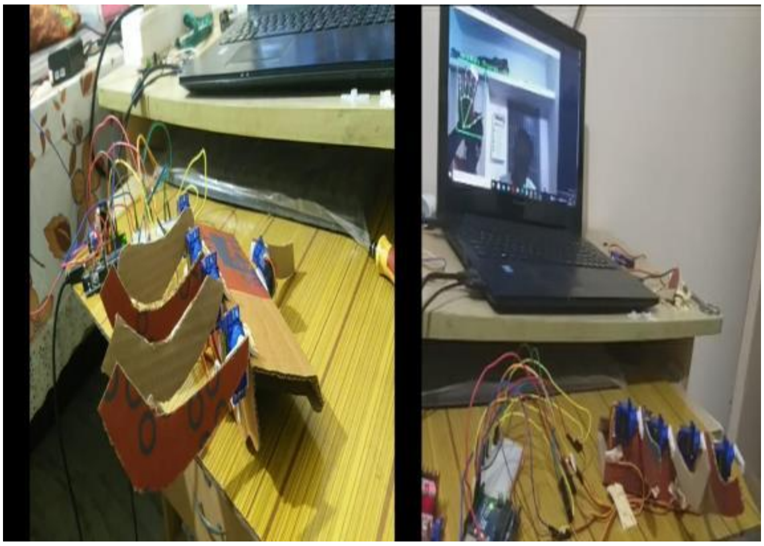

Figure 12. The installation of hardware components

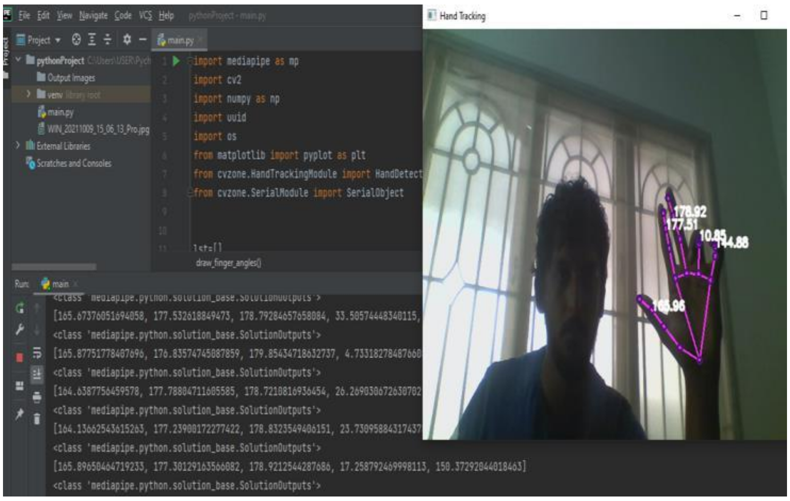

Figure 13. Processed gesture action with angle values

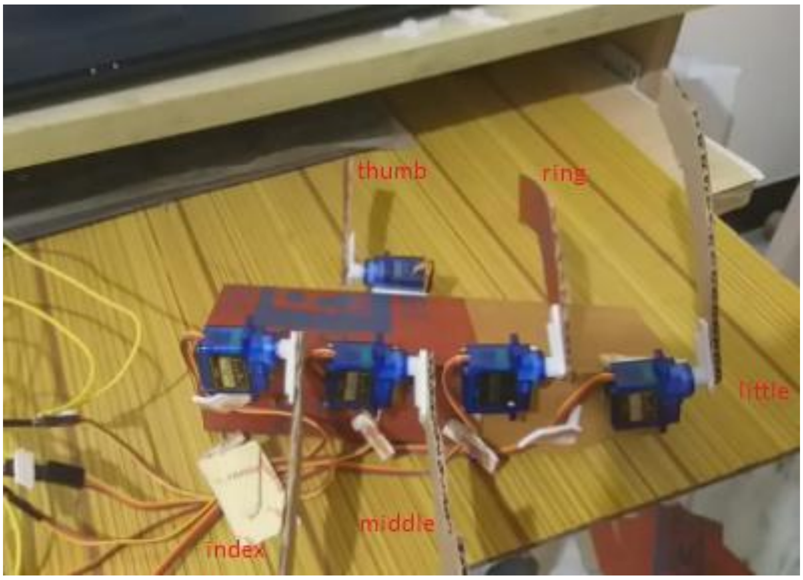

Figure 14. Output from the hardware

Table 1. Output value from the hardware

Input/Output		Fingers				
Trials		Thumb	Index	Middle	Ring	Little
Trial 1	Input Value	180	160	160	170	180
	Acceptable Value	170-180	150-170	150-170	160-180	170-180
	Output Value	178.2	165.8	164.6	175.6	176.4
	Error	-1.8	+5.8	+4.6	+5.6	-3.6
Trial 2	Input Value	160	180	180	10	160
	Acceptable Value	150-170	170-180	170-180	0-20	150-170
	Output Value	165.8	177.3	178.9	17.2	150.3
	Error	-5.8	-2.7	-1.1	+7.2	-9.7
Trial 3	Input Value	150	100	60	40	120
	Acceptable Value	140-160	90-110	50-70	30-50	110-130
	Output Value	155.2	95.8	66.2	42.6	118.8
	Error	-5.2	-4.2	+6.2	+2.6	-1.2

5. RESULT AND DISCUSSION

5.1 Implementation

The code is written in Python and executed in the PyCharm IDE, where a Python script captures and processes video in real-time. The processed data is then sent to an Arduino Uno microcontroller, which uses commands from the Arduino IDE to control the servo motors. The six joints that make up a robot arm are the base, shoulder, elbow, wrist, hand, and gripper. The five joints are considered to actuate the motors in this project. The human hand's five fingers and palm are employed for recognition and detection. Each of the joints in the robot arm is controlled by the five fingers thumb, index, middle, ring, and trim.

The thumb finger actuates the base, the shoulder is actuated by the index finger, the elbow is actuated by the middle finger, the wrist is actuated by the ring finger, and the hand is actuated by the little finger.

The hand is recognized using 21 landmarks in hand. The angle is found using three points on a finger, and the value is then processed and measured. The measured value will be listed and delivered to the Arduino IDE through serial module communication.

The actuation signal pulse for each servo motor will be the processed output for the right hands. All articulated robots with six joints can benefit from this software architecture.

This project's software is mainly developed to connect the user and robot perfectly. The final physical hardware is installed to evaluate the system's functionality, actuation precision, and practicality.

To acquire a video capture object for the camera, call cv2.VideoCapture(). Set up an infinite while loop and read the frames using the object mentioned above's read () function.

To view the frames in the movie, use the cv2.imshow() function. When the user presses a particular key, the loop is broken. Fig. 9 shows Video Capturing.

To extract actionable data from the video captured, use the libraries described in the prior section to process it. The following command is used to create a function that calculates the angle. def draw_finger_angles(image, results, joint_list):

The command below is to extract the coordinates of the predetermined points on MediaPipe hands solutions within the function.

```
a=np.array([hand.landmark[joint[0]].x,hand.landmark[joint[0]].y])
```

The following command is then used to calculate coordinates in radian values.

```
radians = np. arctan2(c[1] - b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
```

The angle is then calculated using the formula below.

```
angle = np.abs(radians*180.0/np.pi)
```

Using the below instruction, this function is used to provide a list of angle data to Arduino.

```
mySerial.sendData(lst)
```

Fig. 10 shows the processing of the Real-time Video.

Data is sent from Python to Arduino through serial communication. It is a serial module that allows access to the serial port. Creating a serial object and assigning configurations to it (port number, baud rate, digits sent per value,

```
mySerial: SerialObject=SerialObject("COM6", 9600, 1)
```

Below is the command used to send the desired data to Arduino. Lists of values transmitted through serial communication are shown in Fig. 11.

```
mySerial.sendData()
```

Hardware must be interfaced with software to view physical movements and actuation. The data transfer from the computer to the Arduino Uno board takes place when the Arduino Uno board is connected to the computer through a USB communication cable. Since the Arduino Uno has six PWM pins, the servo motors may be attached directly to the board. The 9V Battery provides the necessary power to operate the servo motor. Breadboard and jumper wires are used to make the connections. Fig. 12 shows the installation of hardware components.

5.2 Testing

All components are connected during the testing phase, and the Arduino board's USB serial connection is connected to the computer. The PyCharm IDE and Arduino IDE are configured to run the program script. All the joints in the program worked as predicted, and the motor actuation delivered the appropriate output. Fig. 13 shows processed gesture action with angle values. The output from the hardware is shown in Fig. 14. Table 1 shows the test results.

5.3 Application

Human involvement in hazardous environments can be avoided if this project is successfully implemented in the market with high precision. The Gesture-Based Robot Arms Control is intended to be included as an integral part of the project. In the future, a whole arm can be controlled by mimicking a human component using gestures. This project can be implemented in real-world applications like bomb diffusing, segregating hazardous wastes, etc.

6. CONCLUSIONS

This project has implemented and developed Gesture-based robot arms control using OpenCV. Software and hardware interfacing is done, and the desired output is achieved according to the input. This project shows the correct and feasible approach to controlling the robot arm using gestures. More packages have been installed to increase the efficiency of the actions performed.

7. REFERENCES

[1] P. Sihombing, M. B. Rifky, H. Herriyance, and E. Elviwani, "Robotic Arm Controlling Based on Fingers and Hand Gesture," *MECnIT 2020 - International Conference on Mechanical, Electronics, Computer, and Industrial Technology*, pp. 40–45, 2020, doi: 10.1109/MECnIT48290.2020.9166592.

[2] Y. Zhu, Z. Yang, and B. Yuan, "Vision based hand gesture recognition," *Proceedings of International Conference on Service Science, ICSS*, pp. 260–265, 2013, doi: 10.1109/ICSS.2013.40.

[3] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 2015, doi: 10.1016/j.cviu.2015.08.004.

[4] Sakshi Sharma, Shubhank Sharma, and Piyush Yadav, "Design and Implementation of Robotic Hand Control Using Gesture Recognition," *International Journal of Engineering Research and*, vol. V6, no. 04, pp. 302–307, 2017, doi: 10.17577/ijertv6is040352.

[5] N. Nissi Kumar, P. Sai Tharun, K. Prakash, L. M. N. Mythresh, and N. Jyoshna, "Robot ARM with Gesture Control," *Journal of Physics: Conference Series*, vol. 2325, no. 1, 2022, doi: 10.1088/1742-6596/2325/1/012012.